Character Pipeline for Artists

SDK Version 1.0

Contents

Figures

Tables

1 Quickstart

This quickstart is relevant for any development team member interested in how to proceed through the Character Pipeline. It covers:

- Installing the NINTENDO GAMECUBE (GCN) Character Pipeline converter tools.
- Loading and exporting an example scene in 3D Studio MAX Release 3.1.
- Loading and exporting in Maya 3.0.
- Previewing the example scene at runtime.

1.1 Installation

1.1.1 3D Studio MAX

We provide two tools to convert geometry, texture, hierarchy, and animation data from 3D Studio MAX Release 3.1 to the GCN:

- `CPExport.dle` is a 3D Studio MAX Release 3.1 export plug-in. `CPExportD.dle` is the debug version.
- `TexConv.exe` is a texture converter that converts TGA files.

Install the NINTENDO GAMECUBE Character Pipeline SDK (CP SDK), then follow these steps to configure 3D Studio MAX to recognize the Character Pipeline export plug-in:

1. Launch 3D Studio MAX Release 3.1.
2. On the main menu, click Customize > Configure Paths…
3. Click on the "Plug Ins" tab at the top of the dialog box.
4. Click on "Add."
5. Browse to or enter the path `/cp/x86/lib` in the text box. This directory should contain `CPExport.dle` and `CPExportD.dle`.
6. Click on "Use Path."
7. Exit and re-launch 3D Studio MAX to load the plug-in.

To learn how to convert a 3D Studio MAX scene, see section 1.2.1.

1.1.2 Maya

We provide two tools to convert geometry, texture, hierarchy, and animation data from Maya 3.0 to the GCN:

- `CPExport.mll` is a Maya 3.0 export plug-in. `CPExportD.mll` is the debug version.
- `TexConv.exe` is a texture converter that converts TGA files.

Install the CP SDK, then follow these steps to configure Maya to load the converter.

1. Copy `/cp/x86/lib/CPExport.mll` to `Maya3.0/bin/plug-ins`. If you wish to install the debug converter, copy `CPExportD.mll` as well.
2. Copy `/cp/x86/bin/CPExportOptions.mel` to `Maya3.0/scripts/others`.
3. Launch Maya.
4. Click on Window > Settings/Preferences > Plug-in Manager…
5. Click the "loaded" check box next to `CPExport.mll`.

To learn how to convert a Maya scene, refer to section 1.2.2.

1.2 Conversion

1.2.1 3D Studio MAX

We provide several example scenes and characters to demonstrate the Character Pipeline for 3D Studio MAX. Load one of the databases found in `/cp/max` into 3D Studio MAX Release 3.1, then follow these instructions to export the scene:

1. Click File > Export… on the 3D Studio MAX menu.
2. To enable the MAX converter, make sure that "Nintendo CPExport (*.GPL, *.ANM, *.ACT)" is selected next to the "Save as type" drop-down list.
3. Specify the desired file name next to the "File name" edit box.
4. Click "Save" to start the conversion process. (It is not necessary to change any options, as these are saved uniquely for each database.)

You can track the conversion progress in the MAX status bar.

1.2.2 Maya

Currently, we do not provide any example scenes for the Maya converter. Instead, load any Maya scene you wish to export, and follow these instructions:

1. Open the Script Editor so you can view the progress of the converter.
2. Click on the option box for either File > Export All, or File > Export Selection. Only the Maya converter supports the selective export of object.
3. Under "File Type," select "CPExport" in the drop-down list, then select the desired options.
4. Click "Export All" to start the conversion process.

1.3 Previewing

The GCN previewer shows the converted scene at runtime. The previewer runs on the Dolphin Development Hardware (DDH) equipped with a GCN controller.

In order to preview the scene:

1. Make sure these environment variables are defined:
   - `CP_ROOT`: Installed by CP SDK. Location of previewer and texPrev2. For MAC, assumes `dvdroot` is `/cp/cpdata`.
   - `AMCDDKBIN`: Installed by AMC's gameOptix DDK. For HW2 or HW1, supplies DDH commands.
   - `DOLPHIN_PLATFORM`: Installed by GCN SDK, but can be defined manually. Should be one of the following depending on previewing platform: MAC, HW2, or HW1.
2. Start a Cygnus bash shell. Go to the directory where the scene was exported.
3. Execute "preview [name of the scene without three-letter extension]". The scene must be in the current directory.

If the preview platform is a HW2 or HW1 machine, then the `preview` script will automatically execute the appropriate previewer. If the preview platform is a MAC emulator, you must share the `cp` directory onto the Macintosh desktop, then execute the previewer built for the MAC emulator manually.

The script file `preview` can preview GPL, ACT, ANM, or even TPL files. The script file assumes that you want to load the given filename in the order of animation (ANM), actor (ACT), and geometry (GPL). Therefore, it will automatically load the first file to exist of these types in this order. To override the default order, supply the script file with a second argument of "`anm`", "`act`", or "`gpl`".

In addition, you can preview textures in a TPL file by supplying a second argument of "`tpl`" in step 3 above.

If an optional "`stp`" setup file for cameras and lights exists, the previewer will override the default camera and light settings. Currently, only 3D Studio MAX can export a STP file.

The following buttons manipulate the previewer:

- **Stick:** Rotate camera/light.
- **Substick:** Dolly camera/light.
- **Y, A:** Zoom camera/light.
- **X:** Switch to camera, or switch to next camera.
- **Y:** Switch to light, or switch to next light.

- **L:** Reload file and reset camera/light.
- **R:** Shift button (see below).
- **R + Stick:** Roll camera.
- **R + Y:** Change animation rate.
- **R + X:** Next GPL object or ANM sequence.
- **R + B:** Toggle gamma setting.
- **R + A:** Toggle performance bar.
- **Start/Pause:** Pause (Quit for emulator).

You can use the Y button to place and orient lights by controlling them as cameras. The field of view is set wide enough to encompass the angle of the spotlight. For omni-directional lights, the field of view is 90 degrees.

Using 3D Studio MAX

In this chapter, we assume that the reader is already familiar with 3D Studio MAX Release 3.1. We focus on the features and limitations of the MAX converter when exporting a scene.

1.4 Geometry

The MAX converter supports most static geometry objects that are convertible to triangles, including Editable Meshes and Patches.

1.5 Texture

The Character Pipeline supports a maximum of one texture per polygon; therefore, it utilizes only a small subset of the full functionality of 3D Studio MAX materials.

1.5.1 Applying a texture

Currently, the texture converter accepts only targa (TGA) files as input, so all bitmaps used in a scene should be in TGA format. The length and width of these textures should not exceed 1024x1024, and each value must be a power of 2.

The texture converter offers two types of materials to apply textures to geometry objects:

- Standard – for geometry that uses at most one texture.
- Multi/Sub-Objects – for geometry that uses more than one texture (for different groups of polygons).

Follow these steps to associate a texture with a material:

1. Open the Material Editor and select a material.
2. Click on the button to the right of the Diffuse Color selector.
3. Select “Bitmap” from the Material/Map browser and select the desired targa file.
4. In the “Coordinates” panel, select “Explicit Map Channel” for mapping. The texture will tile (or wrap) by default; however, you can clamp the texture by de-selecting both “Mirror” and “Tile.” Mirroring textures is not supported. You also have the freedom to place textures by modifying “Offset,” “Tiling,” and the W “Angle.”
5. Apply the texture to geometry by dragging the appropriate material onto the desired geometry object. (To view the texture in the viewport, click on the “Show Map in Viewport” button.)

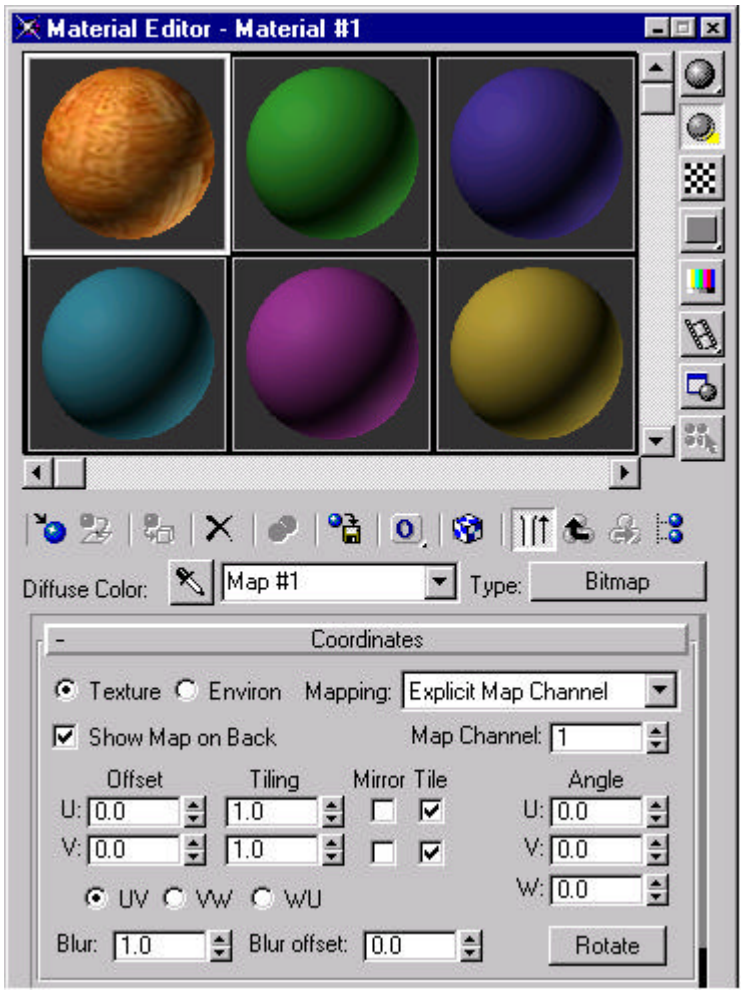

Figure 1 Material Editor with bitmap

1.5.2 Setting texture output format

You can control the output format of the textures by appending suffixes to the bitmap filenames, as shown in Table 1. For example, using the filename "`wood_cmpr.tga`" in 3D Studio MAX will output that texture in `GX_TF_CMPR` (or `S3TC`) compressed format.

Target GCN Texture Format	Filename Rule
GX_TF_I4	*_i4.tga
GX_TF_I8	*_i8.tga
GX_TF_IA4	*_ia4.tga
GX_TF_IA8	*_ia8.tga
GX_TF_RGB565	*_rgb565.tga
GX_TF_RGB5A3	*_rgb5a3.tga
GX_TF_RGBA8	*_rgba8.tga
GX_TF_CMPR	*_cmpr.tga
GX_TF_CI8	*_ci8.tga

Table 1 Texture file naming rules

If these naming rules are not followed in texture filenames, the converter defaults to two texture formats, `RGB565` or `RGB5A3`, depending on whether or not the texture has an alpha component. If the "Auto S3 Texture Cmpr" feature is enabled (see Figure 10), then the converter will default to the compressed texture format. You can verify the output format after conversion by looking at the texture conversion script (TCS) file. Note that only a 1-bit alpha channel is supported when using the compressed `S3TC` format.

Ideally, geometry modeled in 3D Studio MAX should appear with accurate textures on the GCN hardware; however, the texture converter does not understand intensity (`I`) or intensity alpha (`IA`) textures in the TGA file. Therefore, in order to depict `I` and `IA` formats accurately, you should create RGB or RGBA textures while making sure that R=G=B. Furthermore, the `TGA` filename should carry the suffix of the desired intensity format (as shown in the first four rows of Table 1).

1.5.3 Vertex alpha

3D Studio MAX does not handle vertex alpha, which is rather unfortunate because utilizing vertex alpha rather than texture alpha can be an essential space-saving technique, and it can give models a less repetitive, more organic look. We have implemented a workaround to compensate.

Since 3D Studio MAX does not often use the w component of UVW texture coordinates, we can use this as the vertex alpha. Please note, however, that such vertex alphas will be present for an object only when a texture (or UVW Map) is applied to the object.

Apply a material with a bitmap texture to your geometry by following the directions in section 1.5.1, then follow these steps to encode vertex alphas for the object in 3D Studio MAX:

1. Apply a "UVW Map" modifier, then place the texture on the geometry as desired.
2. Use the "Unwrap UVW Map" modifier, then click "Edit…" to edit the texture coordinates.
3. Modify the w coordinate to encode the alpha value (0.0 being fully transparent to 1.0 being fully opaque).
4. Go the object's user-defined properties. Right-click on the object, click on "Properties…", then click the "User Defined" tab. Type "`VertexAlpha=TRUE`" in the window.
5. When exporting, make sure that the "Texture" option is enabled in the user interface.

Figure 2 demonstrates a simple use of vertex alpha with two geometry layers to mimic multitexturing. You can see this technique used extensively in the `Crtyard` and `Knoll` databases. Since the MAX converter does not export multiple textures per polygon, these two databases use vertex alpha quite freely to mimic multitexturing; however, please note that this is not optimal for the hardware because it requires twice the amount of geometry. For more information, check out the `readme` file in the vertex alpha test case in `/cp/max/test/VtxAlpha`.

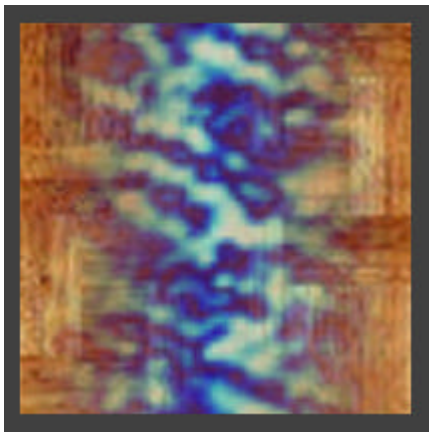

Figure 2 Example of vertex alpha

1.6 Hierarchy

The MAX converter supports the linking of geometry in parent-child relationships by using the "Select and Link" button in the main toolbar. With the Character Studio 2.2 Biped plug-in, you can use this same method to link geometry objects to bones or to segments in a biped. However, remember not to enable any kind of Inverse Kinematics (IK) controller (e.g., when creating bones, de-select the default option of "Assign to Children" in the "IK Controller" panel).

1.6.1 Instancing

Instancing is a simple and useful way to save space. Many instances of the same geometry object can exist in a scene when only one copy is exported. To create an instanced object, hold down the Shift key and translate the geometry object, making sure to enable "Instance" or "Reference." Resetting a transform (a.k.a. "Xform" in 3D Studio MAX) or collapsing the modifier stack will break instancing, so we recommend that you complete the model and perhaps collapse the modifier stack before instancing.

When using the MAX converter, all instances share their parent's material, even though 3D Studio MAX allows a parent and child instance to have different materials (and thus different textures). For consistency, it is a good idea to make sure that the material on the instance is the same material as its parent.

1.6.2 Display order priority

To display transparent objects correctly, objects should be depth-sorted at runtime to determine proper display order. Since the MAX converter does not support depth-sorting, objects can be sorted before runtime to minimize the improper display of transparent objects. This alternate method works particularly well for terrain that uses vertex alpha, like the `Knoll` database, since the camera will always be above the terrain. The `Knoll` terrain uses multiple

layers of transparent geometry for a more compelling, organic look, so the layers on the bottom must be drawn before the layers on the top.

Follow these steps in 3D Studio MAX to specify the order in which geometry objects should be drawn:

1. Go to the object's user-defined properties. Right-click on the object, click on "Properties…", then click the "User Defined" tab.
2. Type "`DrawPriority=`", followed by a number between 1 and 254 (inclusive), on its own line.

Objects in the scene will be rendered in ascending display priority in the GCN previewer. For example, if object "Box" has a display priority of 10, and object "Transparent Sphere" has a display priority of 20, then "Transparent Sphere" will always be drawn after "Box." Objects with the same display priority will be drawn in some arbitrary order. All objects have a default display priority of 0; all non-transparent objects should use the default display priority, while transparent objects should use an appropriate display priority between 1 and 254.

Figure 3 demonstrates the use of display priority. The left branch does not display correctly since it always drawn **before** the background object, while the right branch does display correctly since it is always drawn **after** the background object.

Figure 3 Example of display priority

1.6.3 Biped tail re-parenting

In Character Studio 2.2, the tail bone in a biped character is unique because it does not conform to the hierarchy and animation system in the Character Pipeline. Normally, hierarchy bones are connected end-to-end; this is not true for biped tails—the tail bone is not connected to the end of the spine. One simple solution is to make the tail bone a child of the biped pelvis bone. Since it is not possible to re-parent bones for a biped character in 3D Studio MAX, we must implement this solution in the MAX converter so that the exported animation will be faithful to the 3D Studio MAX original. To start, follow these steps in 3D Studio MAX:

1. Right-click on the tail bone, click on "Properties…", then click the "User Defined" tab.
2. Type "`BipedBoneParent=`[Name of the biped pelvis bone]".

For example, in `/cp/max/SndShrew`, the SandShrew's tail has the user-defined property "`BipedBoneParent=SandshrewBip_Pelvis`" (note that the underscore is necessary because 3D Studio MAX extracts strings without spaces only). Hierarchy and rotation animation information will be corrected for the new parent-child relationship.

1.7 Animation

Although 3D Studio MAX offers many different ways to animate a scene, the MAX converter supports only keyframed animation using a Position/Rotation/Scale (PRS) controller, and biped animation. Moreover, only linear, Bezier (Smooth), or Tension/Continuity/Bias (TCB) interpolation methods should be used when keyframing PRS. There are several restrictions to keep in mind when exporting animation:

- A beginning and ending keyframe must exist during the animation range for each object.
- For any animated geometry object, keyframes must exist at the same times for position, rotation, and scale.
- When animating biped, you should use only default tension, continuity, bias, and ease parameters.
- When using Bezier interpolation or Smooth rotation, the only tested tangent types are Smooth or Custom.
- No IK controllers are supported, even when using a biped.

1.8 Skinning and stitching

If you have Character Studio 2.2 installed, you can use the Physique plug-in to create skinned or stitched characters. Here is a quick definition of terms:

- Skinning allows a vertex to be weighted and blended between more than one bone.
- Stitching allows a vertex to be rigidly attached to only one bone. Stitching is a subset of skinning.

Assuming that you are familiar with Physique, follow these instructions to skin or stitch a geometry object properly for Character Pipeline export:

1. Apply a “Physique” modifier to the geometry object.
2. Click on the “Attach to Node” button and select a bone or biped segment on which to attach the Physique.
3. The “Physique Initialization” dialog box will pop up. In the “Vertex – Link Assignment” panel, “Rigid” **must** be selected. For stitching, select “No Blending” as the option in the “Blending Between Links” drop-down list. For skinning, you can select up to “N Links”. You may wish to create envelopes if you like.
4. Close the panel to complete the initialization.

When you are ready to start assigning weights to vertices, go to the “Physique” modifier in the stack. Select “Vertex” on the “Sub-Object” drop-down list. When assigning vertices to a bone, be aware of the color of the vertex. In order to enforce rigid blending, the only supported color for vertices is green.

Currently, the converter supports only one object with a “Physique” modifier per MAX scene.

1.9 Cameras and lights

Currently, only the 3D Studio MAX CPExport converter has the ability to export cameras and lights pre-defined specifically for the GCN previewer into an STP setup file. There are two types of cameras, differing only in their controls:

- **Free:** Rotates around its own origin
- **Target:** Rotates around target origin

There are only a few restrictions on exporting cameras from 3D Studio MAX. Due to current limitations in previewer controls, the position of the target of a target camera must be at the world origin. For cameras, the “Clip Manually” check box should be enabled in the camera modifier because the GCN always uses camera clipping planes. Then set the settings for “Near Clip” (which must be greater than zero) and “Far Clip” as desired.

Other than the exceptions just noted, you can place and orient cameras however you wish in your scene. You may export as many cameras as desired, and toggle through them in the GCN previewer.

Three types of lights are supported in the CPExport plug-in for MAX:

- **Omni:** Omni-directional light
- **Free Spot:** Local spotlight with free camera-like controls
- **Target Spot:** Local spotlight with target camera-like controls

The GCN hardware only supports local spotlights, so in order to get a direct light effect, place the spotlight far way (so that the light rays are essentially parallel). As is the case with target cameras, the target of the target spotlight must also reside at the origin.

Lighting parameters are trickier. The light color may be set as desired, but since 3D Studio MAX has a different method of lighting than the GCN hardware, parameters are only estimated during conversion.

Under "Spotlight Parameters" in the light modifier, the hotspot and falloff parameters determine the angle attenuation for local spotlights. If the hotspot value is at least 90 percent of the falloff value, the light is considered flat; otherwise, the brightness of light depending on the angle will have a cosine influence. For more technical information, please consult the Dolphin Reference Manual page for `GXInitLightSpot` in `/DolphinSDK1.0/man/index.html`.

Under "Attenuation Parameters", the "Far Attenuation" group is the only value used to estimate the distance attenuation at conversion. If the "Use" check box is enabled, the "End" number will be used to estimate the distance attenuation; otherwise, distance attenuation is not enabled. You may have to tweak the "End" number until the GCN previewer lights your scene as desired.

You can export up to eight lights per scene. If a light is off in the scene, it will not be exported.

2 Using Maya

In this chapter, we assume that the reader is already familiar with Maya 3.0. We focus on the features and limitations of the Maya converter when exporting a scene. As of this writing, the Maya converter is still being matured for game developer use. If you come across any problems, please inform us.

2.1 Geometry

The Maya converter supports most static geometry objects that can be converted to polygons, including NURBS surfaces.

The GCN previewer culls back-facing geometry, although by default, the Maya viewport does not. You may wish to turn this feature on by enabling Shading > Back Face Culling. For NURBS surfaces, you may have to make them face in the opposite direction by using the Edit Surfaces > Reverse Surface Direction utility.

2.1.1 Vertex color

For polygons, vertex colors, not the diffuse material color, will always be exported from Maya. This can cause confusion since the default viewport display shows the material color; to enable vertex colors in the viewport, follow these steps:

1. Click Display > Custom Polygon Display (option box).
2. Make sure that “Color in Shaded Display” is checked.
3. To apply vertex colors, go to the Edit Polygons > Colors popup and use the “Apply Vertex Color” utility on an object or vertex component level. You can also click Window > General Editors > Component Editor… for selected vertices.

Since vertex color is always exported, it will always be multiplied with the texture color if a texture exists. To ensure that only texture color is used, make the appropriate vertex colors white. Be careful—because default vertex colors are black, all polygons will be black upon export if vertex colors are not changed.

For NURBS surfaces, the vertex colors are exported differently because NURBS do not have any concept of vertices. Vertex color will be assigned upon export based solely on the material color. However, if a texture is applied to a NURBS surface, the vertex colors will be white at export so that only the texture colors will be used, even though the material color for the NURBS surface is black.

Vertex alpha is not supported in the Maya converter.

2.1.2 NURBS tessellation

You can explicitly control the tessellation of NURBS surfaces for export. Please follow these steps:

1. Select a NURBS surface.
2. Open the Attribute Editor (type Ctrl-A).
3. Open the “Tessellation” group and click on “Display Render Tessellation.”
4. Open the “Explicit Tessellation Attributes” group and click the “Explicit Tessellation Attributes” check box.
5. Open the “Primary Tessellation Attributes” group and make sure that both the “Mode U” and “Mode V” drop-down lists contain “Per Span # of Isoparms.” Only this selection reliably tessellates NURBS surfaces for export.

6. Modify the “Number U” and “Number V” sliders as desired. For certain NURBS surfaces (like spheres), make sure that both values are at least 2. If either value is 1, then the tessellation may not export as desired or shown in the viewport.

The tessellation will be displayed both in the viewport and under the “Triangle Count” integer box in the “Tessellation” group. This number will be correct for export only if the “Explicit Tessellation Attributes” check box is checked. Remember that only the parameters under “Primary Tessellation Attributes” will be used—even though the group is disabled—if the “Explicit Tessellation Attributes” check box is not checked. No other parameters affect the tessellation for export, even those in the “Secondary Tessellation Attributes” group (for example, the “Smooth Edge” check box should not be checked since it has no effect for the Maya converter).

To verify tessellation of NURBS surfaces, check the triangle count in the statistics file written by the C3 library (it may be helpful to export only triangles, without optimization to triangle strips and fans).

2.2 Texture

The Character Pipeline supports a maximum of one texture per polygon and the Maya converter does not export shaders, so only a minor subset of Maya shaders is utilized to specify textures.

2.2.1 Applying a texture

Currently, the texture converter accepts only targa (`.tga`) files as input, so all bitmaps used in a scene should be in TGA format. The length and width of these textures should not exceed 1024x1024, and each value must be a power of 2.

You can use any Maya shader (including Lambert, Blinn, or Phong) that supports the color channel’s ability to be mapped by a file texture. Follow these steps to associate a texture with a material:

1. Open the Attribute Editor for the shader you wish to use.
2. Click the map button next to “Color” (shown with a checker icon).
3. In the “Create Render Node” dialog box, click the “Textures” tab, then open the “2D Textures” group.
4. Make sure that the “Normal” radio button is selected, then click the “File” button.
5. Select the desired TGA image next to “Image Name” in the Attribute Editor for the new file.
6. Select the desired texture filter next to “Filter Type.” Only the options “Off,” “Box,” or “Mipmap” are supported.
7. Assign the shader to the geometry in one of two ways: middle-click and drag the shader from the Hypershade panel onto the geometry, or hold down the right mouse button on the shader and select “Assign Shader to Selection.”

The Maya converter supports only normal 2D file textures well. None of the other procedural textures provided by Maya, such as Checker or Water, are supported.

You can modify the texture coordinates by using the many utilities under Edit Polygons > Textures.

Texture coordinates may also be modified by using the “place2dTexture” modifier, as long as the “With New Texture Placement” option is enabled in steps 3 and 4 above. Only the “Repeat UV,” “Offset,” and “Rotate UV” modifiers will affect texture coordinates; however, using these modifiers may incorrectly place the texture relative to the viewport upon export. In this case, you will need to normalize the coordinates between 0 and 1 using Edit Polygons > Textures > Normalize UVs.

The shader transparency channel is not supported in the Maya converter. However, the alpha channel in a texture for the shader color map is exported as long as the output format permits the existence of an alpha channel.

2.2.2 Setting texture output format

You can control the output format of the textures by appending suffixes to the bitmap filenames, as shown in Table 1. For example, using the filename "`wood_cmpr.tga`" in Maya will output that texture in GX_TF_CMPR (or S3TC) compressed format.

If these naming rules are not followed in texture filenames, the converter defaults to two texture formats, RGB565 or RGB5A3, depending on whether or not the TGA texture has an alpha component. If the "Auto S3 Texture Cmpr" feature is enabled (see Figure 10), then the converter will default to the compressed texture format. You can verify the output format after conversion by looking at the texture conversion script (TCS) file. Note that only a 1-bit alpha channel is supported when using the compressed S3TC format.

Ideally, geometry modeled in Maya should appear with accurate textures on the GCN hardware; however, the texture converter does not understand intensity (I) or intensity alpha (IA) textures in the TGA file. Therefore, in order to depict I and IA formats accurately, you should create RGB or RGBA textures while making sure that R=G=B. Furthermore, the TGA filename should carry the suffix of the desired intensity format (as shown in the first four rows of Table 1).

2.3 Hierarchy

The Maya converter supports hierarchies as shown in the Hypergraph, including groups and joints. In any transform object, the Shear and Rotate Axis transformations are not supported (and should be zero). However, as long as the rotation and scale pivots are the same, they will properly affect the exported data. All Euler angle rotations should be in XYZ order, and the "Inherits Transform" check box must be enabled.

If the exported scene has misplaced geometry objects, then you may need to use the Modify > Reset Transformations option liberally for all hierarchy transforms in the scene, including joints. If this does not work, then you may have to resort to using the Modify" > "Freeze Transformations utility.

Be careful when using negative scales, since these may not work properly with the GCN previewer.

2.3.1 Joints

The converter supports only a subset of the Maya joint functionality. Please keep these points in mind when working with joints:

- When creating joints, open the "Tool Settings" dialog box (Skeleton > Joint option box), and make sure that the "Scale Compensate" check box is *not* enabled. You can also do this for existing joints by unchecking the "Segment Scale Compensate" box in the Attribute Editor for joints.
- Also in "Tool Settings," uncheck the "Create IK Handle" box since the Character Pipeline does not currently support Inverse Kinematics (IK).
- When exporting stitched geometry (see section 2.5), the joints used by that geometry need to be visible in the scene or Hypergraph for proper export.
- In the Attribute Editor for joints, the "Joint Orient" option will work with the Maya converter.

2.3.2 Instancing

Instancing geometry objects is simple in Maya—use the Edit > Duplicate utility. To verify if a geometry object has been instanced from the original, check whether it appears in the statistics file. It should not be there, since the shape should not have been extracted.

2.4 Animation

The Maya converter strictly supports keyframed animation of translation, rotation, and scale "Transform Attributes" for transform objects, including joints.

The Character Pipeline has some limitations that restrict the exportable animation from Maya:

- For any object, keyframes must exist at the same times for translation, rotation, and scale tracks. For example, if you are extracting translation and rotation animation for a cube, then keys must exist at the same times for translation (*x*, *y*, *z*), and rotation (*x*, *y*, *z*).
- All *x*, *y*, and *z* parameters will always be exported if any one is animated.

Artists do not need to worry too much about these limitations, since the Maya converter will automatically add the necessary keyframes and animation curves. For efficiency, artists should not animate all three x, y, and z parameters if they are always constant.

2.5 Skinning and stitching

The Character Pipeline supports skinning (the method by which a vertex is blended between multiple bones) and stitching (in which a vertex is rigidly attached to only one bone). Follow these instructions to skin or stitch a geometry object properly.

1. Create the desired joints, and position the geometry to be stitched appropriately.
2. Select the geometry, hold down the Shift key, then select the desired root of the joint hierarchy.
3. With the Animation menu set, open the option box for Skin > Bind Skin > Smooth Bind.
4. In the "Smooth Bind Options" dialog box, set "Max Influences" as desired. For stitching, this number should be 1. For skinning, the runtime skinning library can handle an arbitrary number of influences.
5. Click on "Bind Skin" or "Apply" to bind the skin.

For stitching, the Skin > Bind Skin > Rigid Bind tool may seem intuitively like the more appropriate tool. However, the implementation within Maya is very different and is not supported by the converter. In any case, "Smooth Bind" can handle both skinning and stitching cases.

You can check the weights of vertices by opening the Window > General Editors > Component Editor… dialog box, selecting the "SkinClusters" tab, then selecting the desired vertices in the viewport.

NOTE: Currently, the Maya converter supports only one skinned or stitched geometry object per export. This converter has not been thoroughly tested with skinning or stitching export, and thus may not export as desired.

3 Using the converters

This chapter illustrates the basics of how to use either the MAX or Maya converters to export a scene, since the options are similar.

3.1 Options

The conversion process is as simple as setting the desired options and pressing "OK." In the MAX converter only, options are saved automatically each time you save a MAX scene to a file, so you will not have to reset the same options the next time you load the saved scene.

3.1.1 Output

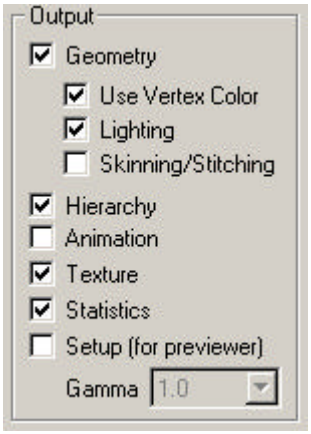

Figure 4 Output

Either converter can export up to seven types of files, each of which bears a three-letter extension according to its purpose:

- `ANM` files describe the animation.
- `ACT` files describe the hierarchy.
- `GPL` files describe the geometry.
- `TCS` files describe how textures should be converted into a `TPL` file.
- `TXT` files contain geometry statistics.
- `SKN` file contain skinning information to supplement the `GPL` file.
- `STP` file contains setup information for the GCN previewer (i.e. cameras and lights).

You can control whether or not to output any of these files by using the check boxes in the "Output" group.

For the setup file, only the 3D Studio MAX `CPExport` plug-in can export this type of file. Currently, the setup file includes information about user-defined cameras and lights for the GCN previewer. Also, the gamma setting can be set as desired with the drop-down box.

NOTE: A dependency exists between GPL and ACT file outputs. The converter performs optimization on the exported hierarchy structure to remove unnecessary bones. A bone is deemed unnecessary if no geometry uses it, so

to make this determination requires geometry output as well. If the converter output an ACT hierarchy file without the relevant GPL geometry file, all bones might end up pruned because they were deemed unnecessary. Therefore, in order to prevent useless cases, the converter does not optimize the hierarchy structure if geometry is not output.

3.1.1.1 Using vertex color

In the MAX converter only, you will also notice that there is a "Use Vertex Color" check box under "Geometry." If this option is checked, the MAX converter will assign colors for each vertex in the scene based on the vertex colors assigned in 3D Studio MAX. Otherwise, the color for each vertex will be the diffuse color of the material assigned to it. Please be aware that the GCN previewer always blends texture colors and vertex colors via multiplication per pixel. So if you would like to make a triangle face show the exact colors of its texture, make sure that the exported vertex colors of the face are white.

3.1.1.2 Lighting

If this option is checked, the converter will export normals for lighting. It will also enable the option for ambient percentage, which is used in dynamic lighting to set the ambient color channel as a percentage of its vertex color in the GCN previewer. Therefore, depending on the intensity of the light, the intended color result will be a percentage value between the ambient percentage and 100 percent of the vertex color. For information on exporting and quantizing normals, see sections 3.2 and 3.3.

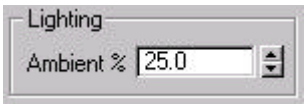

Figure 5 Ambient percentage

3.1.1.3 Skinning/Stitching

Enabling the "Skinning/Stitching" option exports vertex weights for skinning or stitching. For stitching, the converter can put all of the information in the GPL file, but with skinning, the converter will create an SKN file to supplement the GPL file for use in the CPU skinning runtime code. We provide a 3D Studio MAX example of a stitched character in `/cp/max/SndShrew`, and a skinned character in `/cp/max/Zebra`.

3.2 Primitive conversion

Figure 6 Primitive conversion

By default, the converter exports triangle primitives in a scene as triangle strips and fans in order to optimize space and speed for the hardware. You can also export a scene by using triangles entirely.

3.2.1 Viewing triangle strips and fans

Since “tweaking” the geometry can cause inefficiencies in triangle stripping, artists will need visual feedback to fix problematic areas in the geometry and increase the number and length of triangle strips and fans. If the “View Strips/Fans” option is enabled, the converter exports the wireframe of the model with the triangle primitives colored according to their type:

- **Triangles**: red.
- **Quads**: blue.
- **Triangle strips and fans**: random colors.

Stripping efficiency may also be measured quantitatively by calculating the number of vertices per triangle; this information is provided at the bottom of the statistics file. For example, if all the triangle primitives in the model are quads, then the number of vertices per triangle would be two. The lower the number of vertices per triangle, the better the model converts into strips and fans.

NOTE: When enabling the option to view triangle strips and fans, the option to remove duplicate vertex colors per object will not be enabled in the “Compression” group.

3.3 Quantization

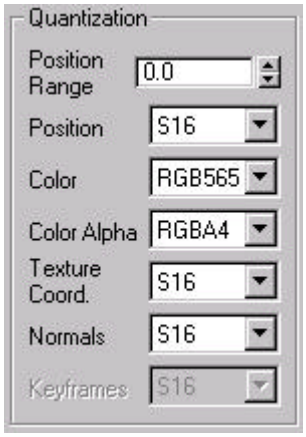

Figure 7 Quantization

Vertex attributes can be quantized to a fixed point format in order to save space. Fixed point numbers use a power of 2 scale, and the converter optimally selects the scale to maximize precision. Below is a list of fixed point GCN formats:

- U8 8-bit unsigned fixed point.
- S8 8-bit signed fixed point.
- U16 16-bit unsigned fixed point.
- S16 16-bit signed fixed point.
- FLOAT 32-bit floating point.

For vertex colors, objects will be automatically quantized, with or without the alpha component, into the following vertex color formats:

Vertex Color without Alpha	Vertex Color with Alpha
RGB565 16 bits (default)	RGBA4 16 bits (default)
RGB8 24 bits	RGBA6 24 bits
RGBX8 32 bits	RGBA8 32 bits

Table 2 Vertex color quantization

The position range is used as a "ruler" in order to quantize positions in the same manner. This creates a fixed global grid of positions among various scenes, as long as these scenes are exported with the same position range. In order to guarantee this fixed global grid, you should make sure that all pivot points (or node transformations) have the same scale and are oriented to each other orthogonally. If the position range is insufficient to encompass all positions, the converter will issue a warning with a suggested minimal position range. For convenience, if the position range is zero, the converter calculates and uses the minimal position range automatically.

NOTE: You do not have quantization control for skinned vertices because these positions and normals will always export to a signed, 16-bit fixed point number. However, you may use the position range option to set the quantization scale for skinned objects.

3.4 Normal table

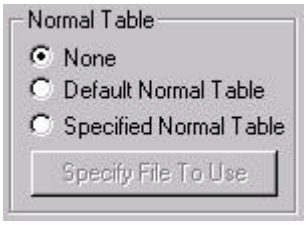

Figure 8 Normal table

Another effective space-saving technique is to utilize a global normal table for all geometry objects to be loaded at runtime, instead of exporting an array of normals per geometry object. The default normal table that we have provided works with the GCN previewer and consists of 252 normals distributed roughly uniformly in all directions. You can specify a different normal table, but this would not be supported by the GCN previewer.

Where better normal precision is necessary, select the "None" option.

3.5 Compression and welding

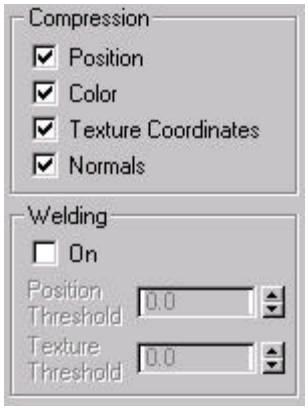

Figure 9 Compression and welding

By default, the converter compresses all position, color, and texture-coordinate arrays to remove duplicates per geometry object. If you would rather not remove duplicates, then uncheck the appropriate box.

Welding is a superset of compression. You may weld two or more distinct positions or texture coordinates when they should be considered the same point or texture coordinate. Welding helps to eliminate very small errors caused by tool imprecision, and may help increase triangle stripping efficiency.

3.6 Texture options

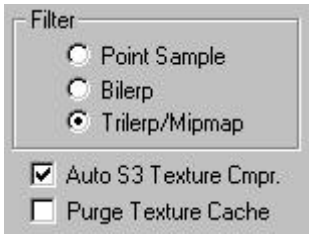

Figure 10 Texture options

In the MAX converter, the texture filter option is global for all textures in the scene. It offers:

- Point sampling.
- Bilinear interpolation.
- Trilinear interpolation with mipmapping.

When enabling the “Trilerp/Mipmap” filter, the width and height of the textures must be a power of 2; since color-indexed textures cannot be mipmapped, the filters for these textures will be bilinear implicitly.

In the Maya converter, texture filters can be set per file texture as described in section 2.2.1.

The “Auto S3 Texture Cmpr” feature is a quick and simple means to make the default texture format compressed, rather than RGB565 or RGB5A3, for all exported textures. However, the texture filename suffix will always be the final output format. Please see section 1.5.2 or 2.2.2 for more information.

We use a texture cache to speed up the conversion process when repeatedly exporting the same scene. To generate a fresh TPL file, enable the "Purge Texture Cache" option so that the texture cache information of the previously-exported scene will be ignored. Also, make sure that a "`C:\temp`" directory exists in your computer as a place to store the cache.

3.7 Finishing up

Once you have set all of your options as desired, click "OK" or "Export" to start the conversion process. In 3D Studio MAX, you can view progress reports in extraction, optimization, and conversion in the status bar. In Maya, you will have to open the Script Editor in order to view the progress of the converter.

Near the end of the operation, both converters call the texture converter (`TexConv`) through the command line to create the TPL file. If there are any errors in texture conversion, they will be noted in the command prompt window that pops up.

If you enabled the option to export statistics, only the MAX converter will display the statistics of the exported GPL file in a pop-up window when conversion is complete. However, you can view the statistics file later by opening the file in the export path suffixed with "`.txt`." The statistics file outlines the specific breakdown size (in bytes) of the GPL file, as well as count and quantization information. The end of the file shows some global summary statistics.

3.8 Example scenes

We have provided some 3D Studio MAX scenes in `/cp/max` that demonstrate of exporting with the MAX converter:

- `Zebra` shows a skinned zebra with pre-set camera angles.
- `Crtyard` shows a Japanese tea garden.
- `Knoll` shows a "Banjo-Kazooie"-like level.
- `Monkey` shows a biped with multiple animation sequences.
- `SndShrew` shows a stitched and animated biped.
- The `Test` directory contains some minor examples that test the `CPExport` converter in 3D Studio MAX. Each subdirectory contains a `readme` file explaining the test, aimed primarily at CG tool developers.

Note that the `Crtyard` and `Knoll` scenes were created in the Character Pipeline before multitexture support; however, these demos achieve a similar effect by using two layers of identical geometry with vertex alpha. Each method has its own tradeoffs. When implementing the two-layer method, the artist must take care that the two layers of geometry are exactly identical in order to prevent flickering (see sections 1.5.3 and 1.6.2 for more information). This method also has the disadvantage of using twice the amount of geometry, but it does not suffer from a halved fill rate.

3.9 Troubleshooting

3.9.1 3D Studio MAX

Image looks fine in 3D Studio MAX, but wrong in the converter; resetting XForm sometimes fixes it

In general, the MAX converter is robust with its basic feature set, however, some features such as mirroring and negative scaling are not well supported. You can fix many problems like these by using the "Reset XForm" utility to reset the transform of geometry objects; if necessary, hierarchy relationships may need to be broken before you reset transforms.

Transparent objects do not display correctly

If a geometry object using texture alpha is not displaying correctly on top of an opaque geometry object, then you should set a display priority for the transparent object (for more information, see section 1.6.2). However, note that display priority may not ensure that transparent objects always display correctly on top of other transparent objects.

Transparent objects have an outline along their edges

The texture in question should be point-sampled by selecting this option upon export, or the texture should be clamped by de-selecting both “Mirror” and “Tile” in the Material Editor for the texture.

Biped footsteps drift in the previewer, but not in 3D Studio MAX

This occurs because bipeds in 3D Studio MAX use IK controllers to keep the feet planted on the ground, but the MAX converter does not support a method for converting IK information. In order to compensate, it is necessary to use more keys; one effective method is to create a key at every frame for the biped, then do a key reduction.

Texture filename paths are stale

When you export a 3D Studio MAX scene with the MAX converter, the textures may not convert properly into a TPL file because the texture paths in 3D Studio MAX are incorrect. The problem arises when the textures used by 3D Studio MAX files (`*.max`) are moved to a different absolute directory path, meaning that the paths to the bitmaps are no longer valid within the MAX file. (This occurred with the targa files in Dolphin Emulator Release e2.5.)

There are several ways to check for this problem:

1. Before conversion, verify the texture paths by clicking on File > Summary Info….
2. Before conversion, perform a Quick Render on the scene. 3D Studio MAX will alert you to any textures that cannot be found.

After conversion, examine the TCS file to verify the texture paths that the MAX converter obtained from 3D Studio MAX.

Besides the painstaking method of fully resetting all of the texture paths in the Material Editor, there are two quick and easy methods to remedy this problem:

1. Keep the bitmap texture maps in the same directory as the MAX file, then save the MAX file using “File > Save…” so that the proper texture paths will be written inside the new MAX file. This works because 3D Studio MAX will search its current directory for any missing bitmaps when textures cannot be found. It may be necessary to save the MAX file with a different filename before conversion by clicking on “File > Save As….”
2. Force 3D Studio MAX to search in certain directories by configuring the bitmap paths. Click on “File > Configure Paths…” and then click on the "Bitmap" tab. Then manually add the directory which contains the textures used in the scene.

When texture paths are correct, a fresh TPL file should result after using the MAX converter.

3.9.2 Maya

Image looks fine in Maya, but wrong in the converter; resetting and/or freezing transforms sometimes fixes it

Maya provides many flexible ways of specifying hierarchy transformations, and since the Character Pipeline supports only a subset of these, the exported scene may not look like the Maya viewport. The source of many of these problems can be fixed by simplifying hierarchy transformations. Use the “Reset Transformations” and/or the “Freeze Transformations” utility.

Textures are not placed correctly

This may occur in some parameters of the place2dTexture modifier. Refer to section 2.2.1 for more information on which parameters affect the exported scene and the need to normalize UV coordinates. Also only shaders with normal 2D file textures mapped in the color channel have been tested to work properly for export.

Geometry objects appear with incorrect or black colors, even if a texture is applied.

For polygons, vertex colors will always be exported, not the material color. By default, Maya assigns the vertex colors to black, and since the GCN previewer always blends texture color and vertex color, geometry may be colored black even if a texture is applied. Refer to section 2.1.1 for more information.

